Hemanshi Marwaha

July 30, 2024

0.1 Introduction

In this analysis, I have explored the dataset containing information on vessel counts and CO2 emissions across three years: 2022, 2023, and 2024 for the month of April. There are **271238 rows** in the dataset. The objective is to identify trends, distributions, and changes in these metrics over time.

0.1.1 Assumptions for the Entire Analysis

- The dataset is assumed to be accurate and comprehensive.
- No missing values significantly impacting the analysis.
- The data is representative of the population and time period it covers.
- External factors affecting vessel counts and emissions are constant or negligible.

[52]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from geopy.geocoders import Nominatim  # Import Nominatim geocoder
```

[53]:
```
import pandas as pd
# Load the dataset from the specific sheet "rawdata"
file_path = '/MyDataset.xlsx'
df = pd.read_excel(file_path, sheet_name='RawData')
```

[54]:
```
df.isnull().sum()
```

[54]:
```
geohash                          0
Qty_vessels_2022                 0
Qty_vessels_2023                 0
Qty_vessels_2024                 0
c02_emissions_2022               0
c02_emissions_2023               0
c02_emissions_2024               0
Pctg_Emissions_2022_Vs_2023      0
Pctg_Emissions_2023_Vs_2024  28216
wkt                              0
dtype: int64
```

This shows that **28216** values in the column *Pctg_Emissions_2023_Vs_2024* are null.

[55]:
```
print(df.info())                          # Display basic information about the␣
 ↪dataset
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 271238 entries, 0 to 271237
Data columns (total 10 columns):
 #   Column                       Non-Null Count   Dtype
---  ------                       --------------   -----
 0   geohash                      271238 non-null  object
 1   Qty_vessels_2022             271238 non-null  int64
 2   Qty_vessels_2023             271238 non-null  int64
 3   Qty_vessels_2024             271238 non-null  int64
 4   c02_emissions_2022           271238 non-null  float64
 5   c02_emissions_2023           271238 non-null  float64
 6   c02_emissions_2024           271238 non-null  float64
 7   Pctg_Emissions_2022_Vs_2023  271238 non-null  float64
 8   Pctg_Emissions_2023_Vs_2024  243022 non-null  float64
 9   wkt                          271238 non-null  object
dtypes: float64(5), int64(3), object(2)
memory usage: 20.7+ MB
None
```

[56]:
```
# Check for duplicate rows
duplicates = df.duplicated().sum()
print("Duplicate Rows:")
print(duplicates)
```

```
Duplicate Rows:
0
```

I did not find any duplicate rows in the given dataset.

There could be several reasons for **28216** values being **0** in the *c02_emissions_2023* column:

Data Collection Methodology: The zeros might indicate areas where no emissions were recorded due to the absence of monitoring equipment or deliberate exclusion in the data collection process.

Measurement Thresholds: Emissions below a certain threshold might be recorded as zero. This could be due to the precision of the measuring instruments or reporting standards.

Seasonal or Operational Shutdowns: Certain locations might have had no emissions during specific periods due to shutdowns of industrial activities, reduced operations, or seasonal closures.

Data Entry Errors: Zeros might be present due to data entry errors, missing data being incorrectly replaced with zeros, or issues in data processing.

Natural Emission Levels: In some regions, natural emissions could inherently be zero or very low, particularly in non-industrial areas or regions with effective emission control measures.

Reporting Standards: The dataset might have a standard where missing or unrecorded data is represented as zero.

0.2 Analyses and Explanations

0.2.1 1. Descriptive Statistics

Methodology: I began my analysis with descriptive statistics to summarize the central tendency, dispersion, and shape of the dataset's distribution.

Explanation: Descriptive statistics offer a simple summary about the sample and the measures. The central tendency (mean, median) gives us an average value, while the dispersion (standard deviation, variance) shows us the spread of the data.

Assumptions: * The data is assumed to be accurate and representative of the population. * No significant outliers are assumed to distort the measures of central tendency.

[57]:
```
df.describe()             # Summary statistics to check for any anomalies
```

[57]:
```
       Qty_vessels_2022  Qty_vessels_2023  Qty_vessels_2024  \
count     271238.000000     271238.000000     271238.000000
mean          14.891792         15.879781         28.158407
std           40.741093         46.031393         63.271073
min            1.000000          1.000000          1.000000
25%            3.000000          3.000000          5.000000
50%            5.000000          6.000000         11.000000
75%           12.000000         12.000000         27.000000
max         1465.000000       1808.000000       2169.000000

       c02_emissions_2022  c02_emissions_2023  c02_emissions_2024  \
count       271238.000000       271238.000000       271238.000000
mean           117.410697          119.056558          229.463580
std            373.262770          381.213865          586.183310
min              0.019140            0.000000            0.000000
25%             11.927710            9.939541           24.356133
50%             31.758911           32.180022           75.098193
75%             92.003956           92.333361          224.560354
max          53060.348430        28327.144030        62885.305390

       Pctg_Emissions_2022_Vs_2023  Pctg_Emissions_2023_Vs_2024
count                271238.000000                 2.430220e+05
mean                     53.505371                 3.634968e+02
std                    1101.908751                 9.247017e+03
min                    -100.000000                -1.000000e+02
25%                     -53.018853                 1.160574e+01
50%                      -6.380406                 1.174063e+02
75%                      54.792911                 2.967791e+02
```

```
max                    282893.989500                   3.852202e+06
```

The mean of the vessels quantity doubled from 2022 to 2024. The same can also be seen for the CO2 emissions.

The max value for CO2 emissions in 2023 is half of what it was in the year 2022.

[58]:
```python
import geopandas as gpd
from shapely import wkt

# Convert WKT to geometry

df['geometry'] = df['wkt'].apply(wkt.loads)
gdf = gpd.GeoDataFrame(df, geometry='geometry')

gdf.head(2)
```

[58]:
```
  geohash  Qty_vessels_2022  Qty_vessels_2023  Qty_vessels_2024  \
0   kdyfp                37                44                63
1   kdzyb                25                28                52

   c02_emissions_2022  c02_emissions_2023  c02_emissions_2024  \
0          191.624862           218.69740          493.144696
1          151.940106           129.90579          560.807735

   Pctg_Emissions_2022_Vs_2023  Pctg_Emissions_2023_Vs_2024  \
0                    14.127884                   125.491797
1                   -14.501975                   331.703417

                                                  wkt  \
0  POLYGON ((32.2998047 -29.1796875, 32.2998047 -…
1  POLYGON ((33.3984375 -28.3447266, 33.3984375 -…

                                             geometry
0  POLYGON ((32.29980 -29.17969, 32.29980 -29.135…
1  POLYGON ((33.39844 -28.34473, 33.39844 -28.300…
```

[59]:
```python
#pip install pandas matplotlib geopandas python-docx
```

[60]:
```python
sns.set()   # Resets to seaborn default style
plt.rcdefaults()  # Resets to matplotlib default style
```

0.2.2 2. Geospatial Analysis

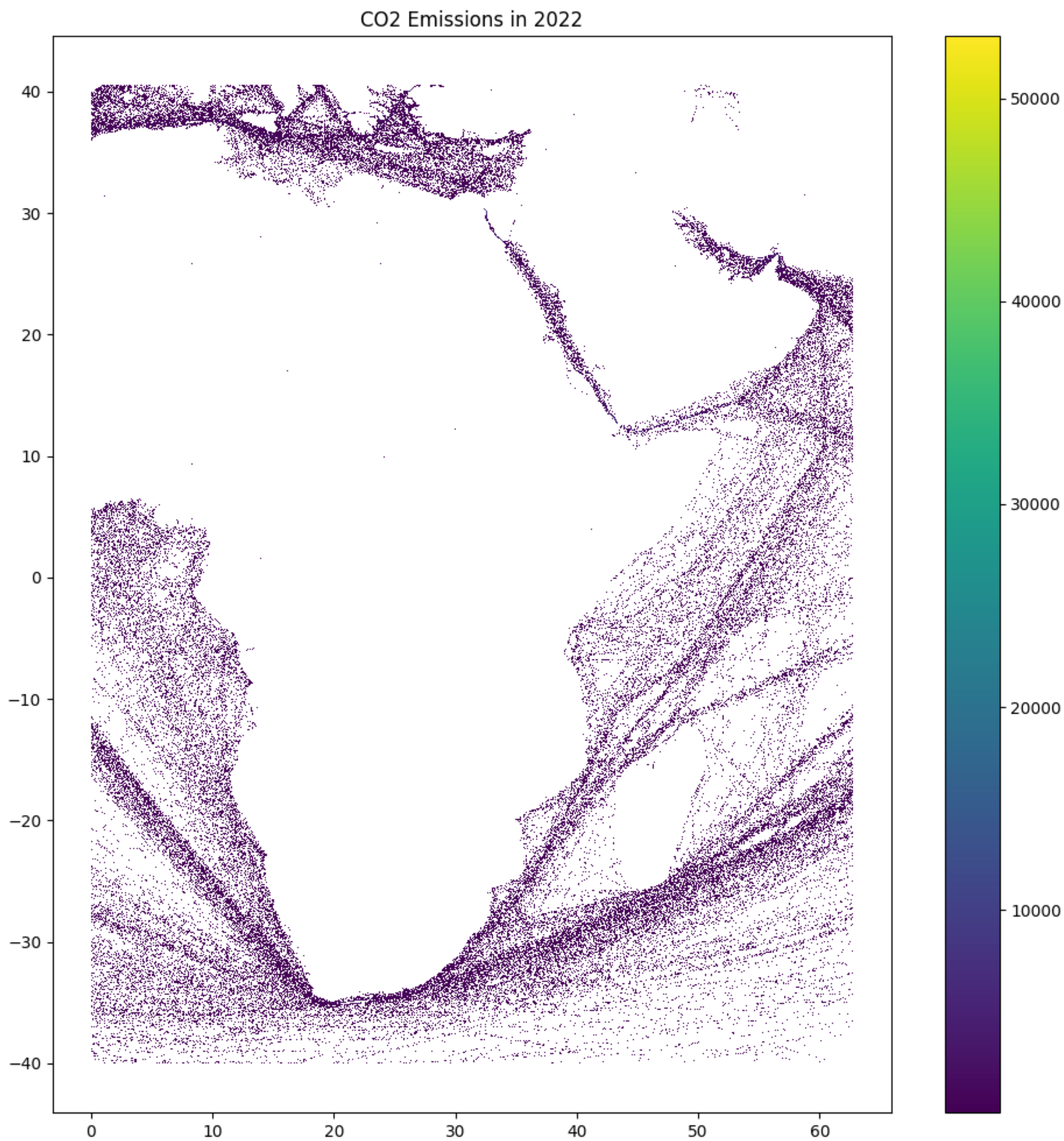
Methodology: We use geospatial analysis to visualize the distribution of vessels and emissions geographically. This involves plotting points on a map based on latitude and longitude.

Explanation: Geospatial analysis helps in identifying geographical patterns and hotspots. By visualizing vessel counts and emissions on a map, we can see if certain areas have higher activity or emissions than others.

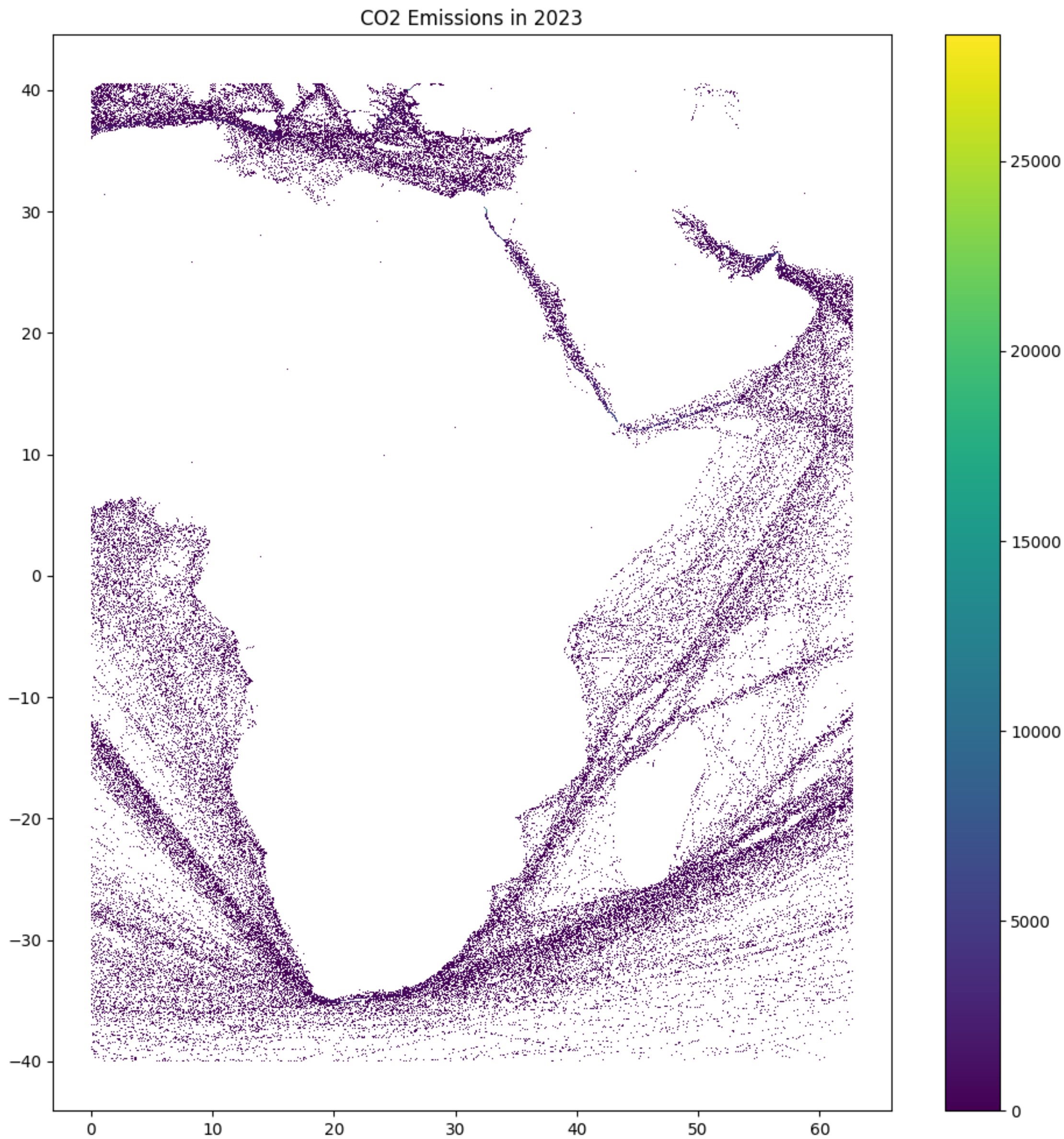
Assumptions:

- The coordinates provided in the dataset are accurate.
- The map used covers the relevant geographical area comprehensively.

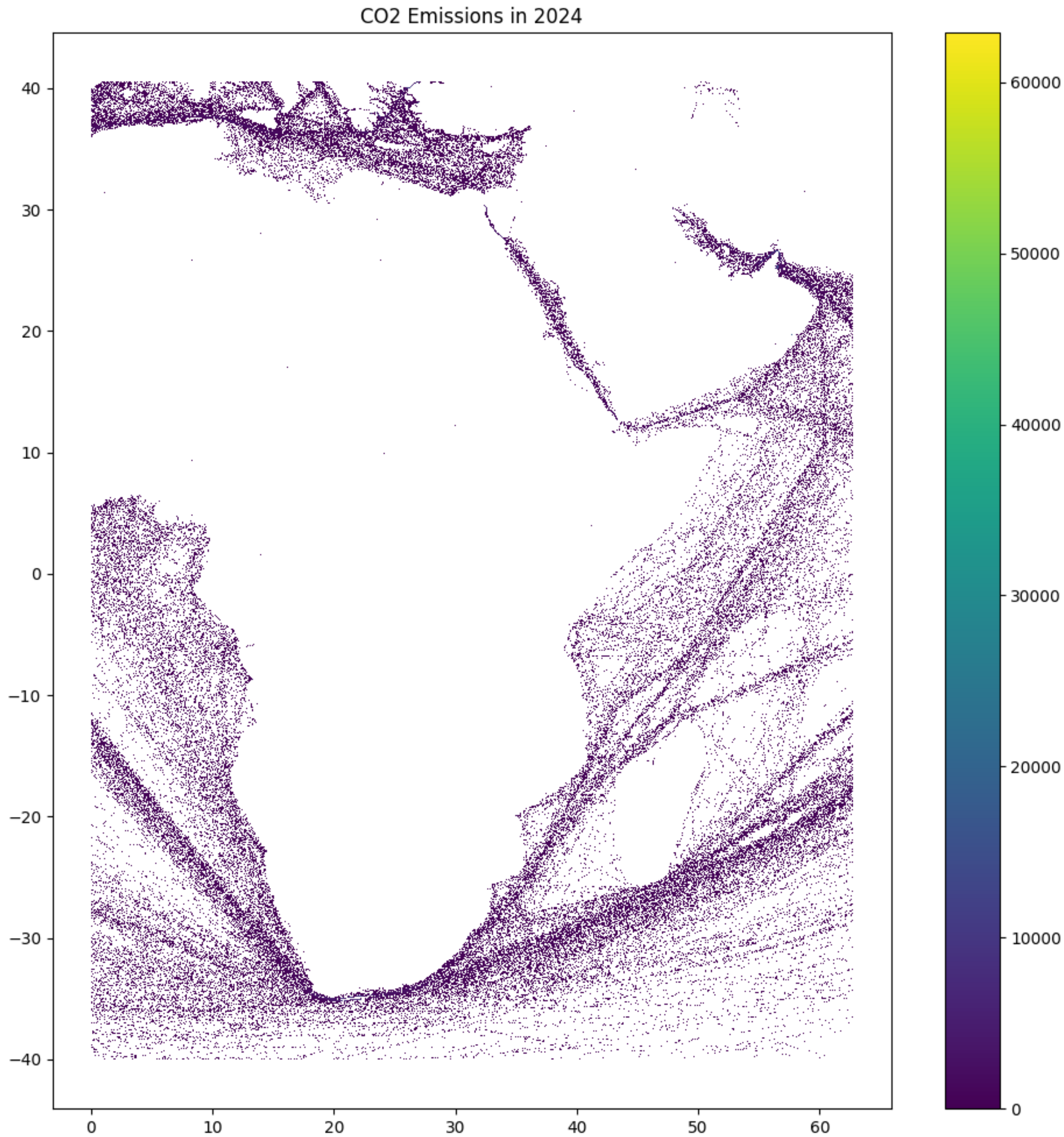
[61]:
```
# Plot geospatial data
gdf.plot(column='c02_emissions_2022', legend=True, figsize=(12, 12))
plt.title('CO2 Emissions in 2022')
plt.show()
```



[62]:
```python
gdf.plot(column='c02_emissions_2023', legend=True, figsize=(12, 12))
plt.title('CO2 Emissions in 2023')
plt.show()
```



[63]:
```python
gdf.plot(column='c02_emissions_2024', legend=True, figsize=(12, 12))
plt.title('CO2 Emissions in 2024')
plt.show()
```



These maps show the geospatial distribution of CO2 emissions over the regions for the years 2022, 2023, and 2024. Each point represents a location where CO2 emissions were recorded, and each map uses a color gradient to represent the intensity of CO2 emissions, with higher values indicated by brighter colors.

There appears to be a reduction in the maximum emission values in 2023 as compared to 2022, as indicated by the color scale, which peaks at a lower value. However, there is a noticeable increase in the maximum emission values in 2024, as the color scale now peaks at a higher value.

The reduction in peak values in 2023 could indicate the effectiveness of emission reduction measures or changes in shipping activity.

The increase in 2024 might suggest a resurgence in shipping activities or a possible relaxation of emission controls.

Comparative Analysis

Consistency: Across all three years, the emission patterns remain consistent.

Trends:

There is a notable decrease in the 'maximum' values of emissions from 2022 to 2023, followed by an increase in 2024.

Overall, the total emissions slightly increased in 2023. However, there is a significant increase in emissions in 2024. This is because of number of vessels increased significantly from 2023 to 2024.

Implications: Understanding these patterns is crucial for policymakers and environmental agencies to target emission reduction efforts effectively. The observed trends highlight the need for sustained and possibly more stringent measures to control CO2 emissions in maritime activities.

0.2.3 3. Correlation Analysis

Methodology: We calculate the correlation matrix to determine the relationships between vessel counts, CO2 emissions, and other relevant variables.

Explanation: Correlation analysis helps in identifying how variables are related. Positive correlation indicates that as one variable increases, the other tends to increase, and vice versa.

Assumptions:

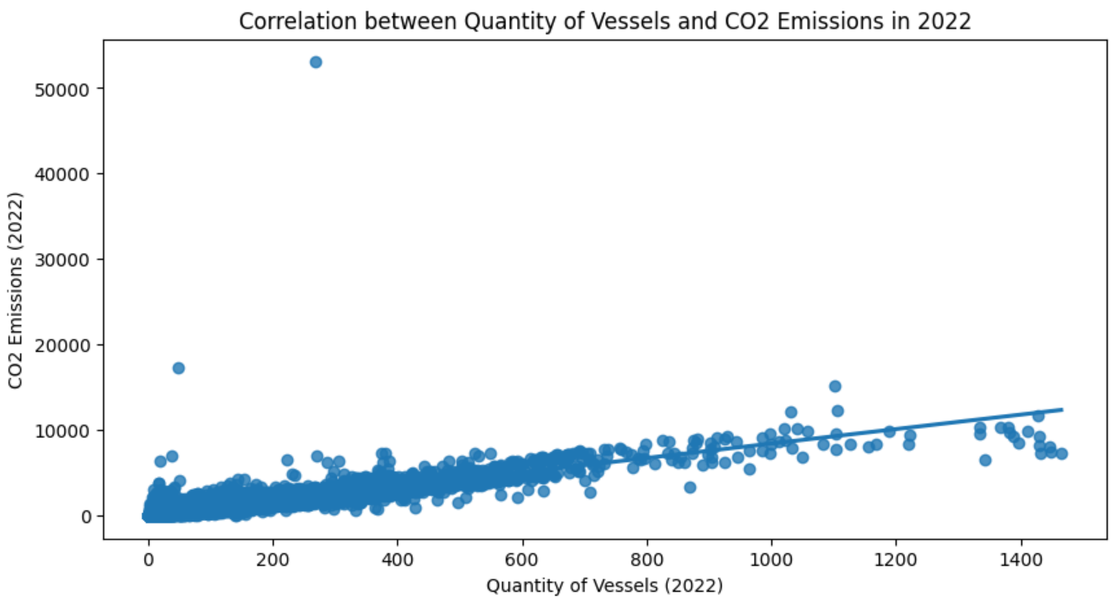
- The relationships between variables are linear.
- The data is free from significant outliers that can skew the correlation.

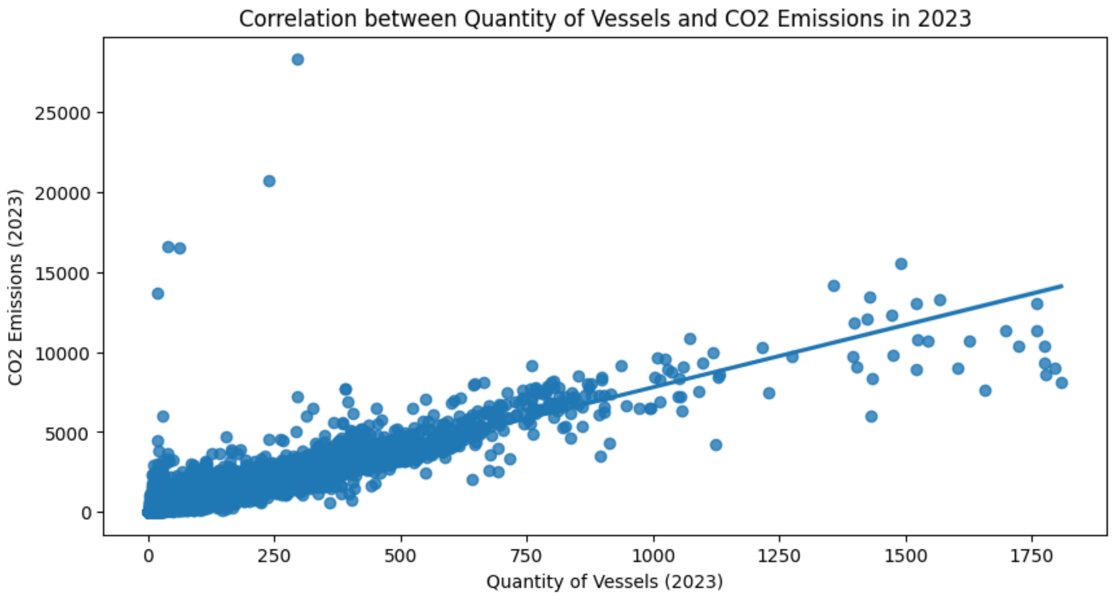
[64]:
```
# Correlation between Qty_vessels and c02_emissions for each year
corr_2022 = df['Qty_vessels_2022'].corr(df['c02_emissions_2022'])
corr_2023 = df['Qty_vessels_2023'].corr(df['c02_emissions_2023'])
corr_2024 = df['Qty_vessels_2024'].corr(df['c02_emissions_2024'])

print(f'Correlation in 2022: {corr_2022}')
print(f'Correlation in 2023: {corr_2023}')
print(f'Correlation in 2024: {corr_2024}')
```

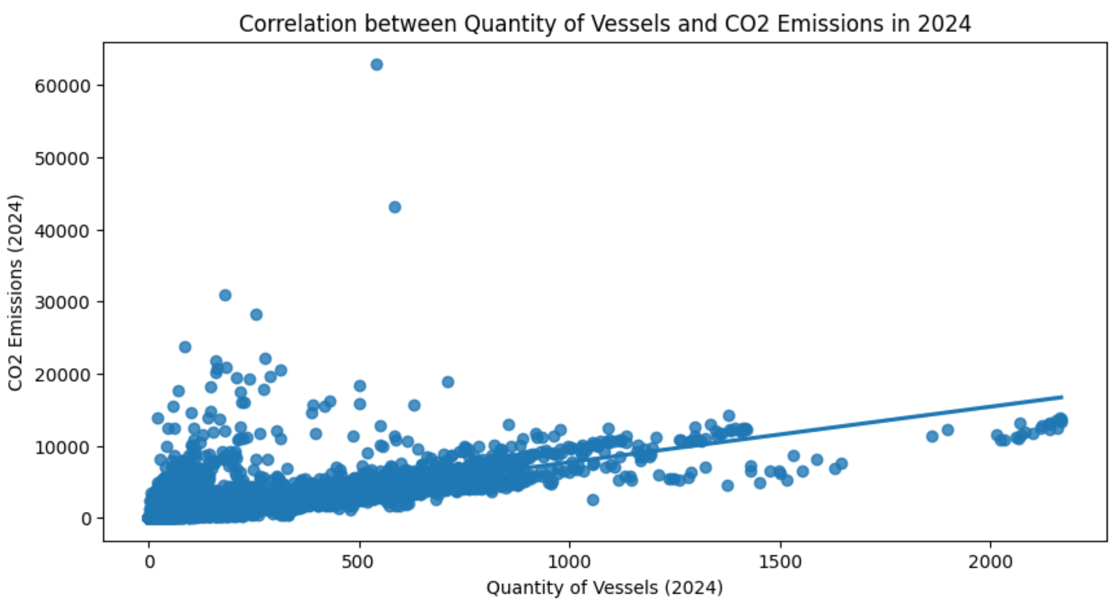
```
Correlation in 2022: 0.9219424341815772
Correlation in 2023: 0.9428580079000095
Correlation in 2024: 0.8303382381875465
```

[65]:
```
# Correlation Plots
plt.figure(figsize=(10, 5))
sns.regplot(x='Qty_vessels_2022', y='c02_emissions_2022', data=df)
plt.xlabel('Quantity of Vessels (2022)')
plt.ylabel('CO2 Emissions (2022)')
plt.title('Correlation between Quantity of Vessels and CO2 Emissions in 2022')
plt.show()

plt.figure(figsize=(10, 5))
```

```
sns.regplot(x='Qty_vessels_2023', y='c02_emissions_2023', data=df)
plt.xlabel('Quantity of Vessels (2023)')
plt.ylabel('CO2 Emissions (2023)')
plt.title('Correlation between Quantity of Vessels and CO2 Emissions in 2023')
plt.show()

plt.figure(figsize=(10, 5))
sns.regplot(x='Qty_vessels_2024', y='c02_emissions_2024', data=df)
plt.xlabel('Quantity of Vessels (2024)')
plt.ylabel('CO2 Emissions (2024)')
plt.title('Correlation between Quantity of Vessels and CO2 Emissions in 2024')
plt.show()
```







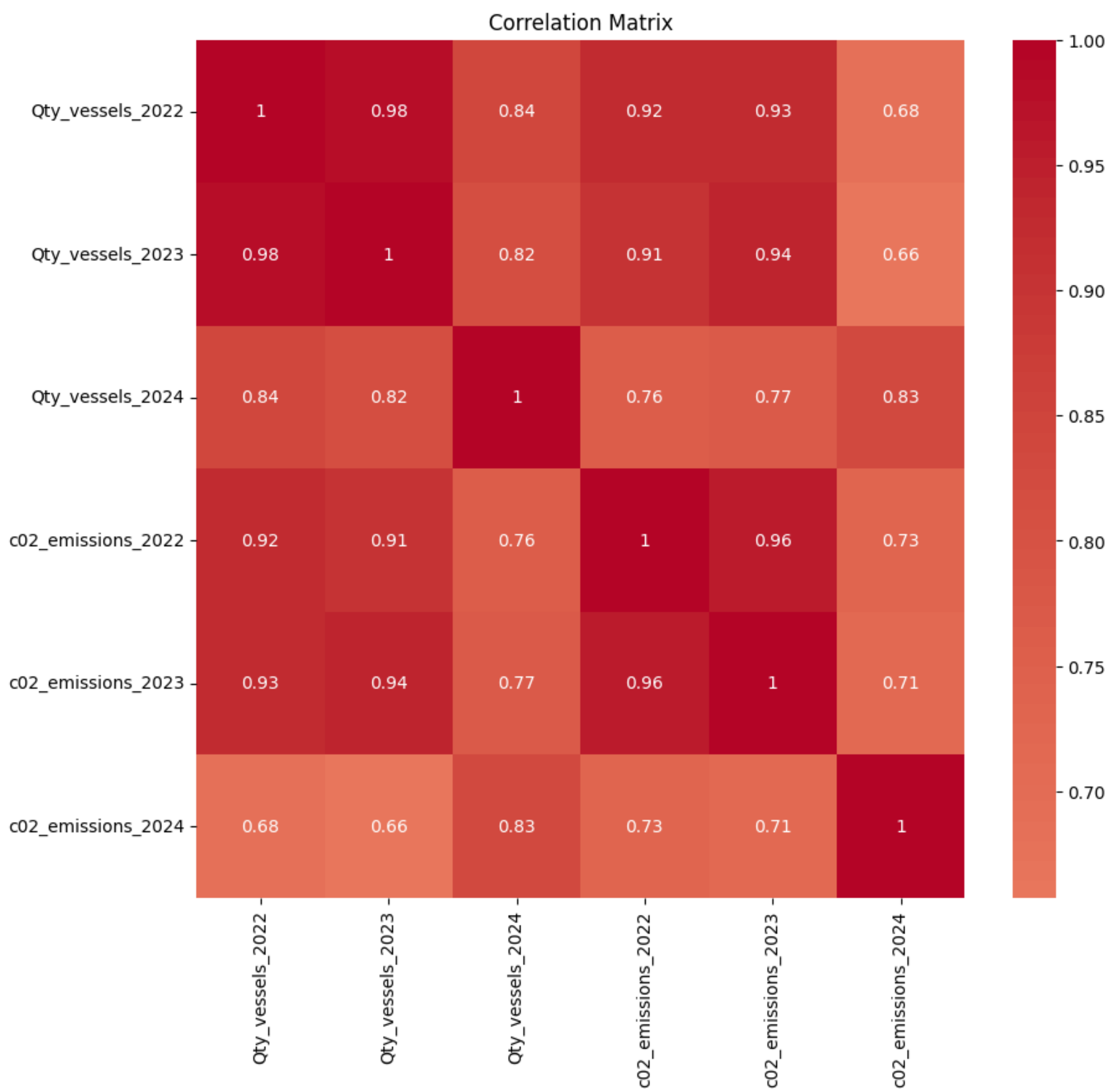
Correlation in 2022 is 0.92. This value suggests a very strong positive relationship between the quantity of vessels and CO emissions in 2022. As the number of vessels increased, CO emissions tended to increase significantly.

Correlation in 2023 is 0.94. This value is even higher, indicating an extremely strong positive relationship between the quantity of vessels and CO emissions in 2023. The trend from the previous year continues, with CO emissions increasing as the number of vessels increases.

Correlation in 2024 is 0.83. Although still strong, this value is slightly lower compared to the previous years. It shows a strong positive relationship, but the association between the number of vessels and CO emissions is not as consistent as in 2022 and 2023. There may be some variability or changes in other factors affecting CO emissions in 2024.

In summary, the high correlation values across these years suggest that there is a strong and consistent relationship between the quantity of vessels and CO emissions, although there is a slight decline in the strength of this relationship in 2024.

[66]:
```
# Calculate correlation matrix

correlation_matrix = df[['Qty_vessels_2022', 'Qty_vessels_2023',␣
 ↪'Qty_vessels_2024',
                          'c02_emissions_2022', 'c02_emissions_2023',␣
 ↪'c02_emissions_2024']].corr()


# Plot heatmap of the correlation matrix
plt.figure(figsize=(10, 9))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', center=0)
plt.title('Correlation Matrix')
plt.show()
```



0.2.4 4. Year-over-Year Change Analysis

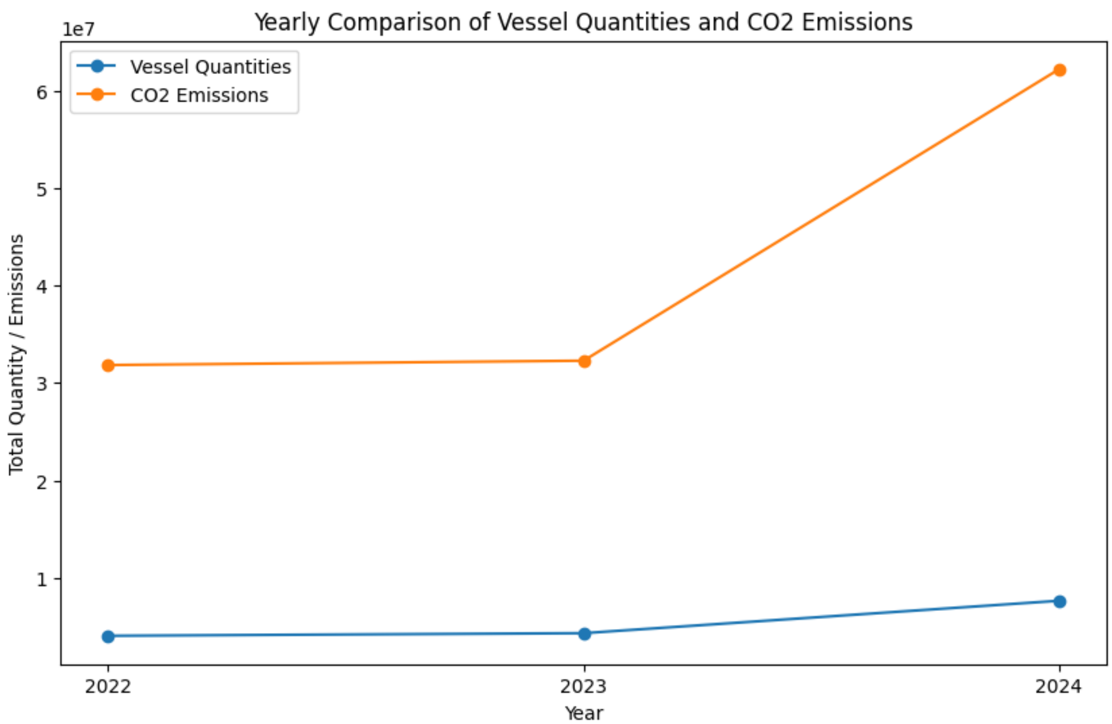
Methodology: We calculate the year-over-year changes in vessel counts and emissions to observe how these metrics have evolved over time.

Explanation: YOY change analysis helps in understanding the growth or reduction in vessel counts and emissions from one year to the next. This is critical for identifying trends and assessing whether interventions are needed.

Assumptions: * The yearly data is consistent and comparable. * External factors affecting the yearly changes are constant or negligible.

[67]:
```
# Compare vessel quantities and CO2 emissions across years
years = ['2022', '2023', '2024']
vessel_quantities = [df[f'Qty_vessels_{year}'].sum() for year in years]
```

```
emissions = [df[f'cO2_emissions_{year}'].sum() for year in years]

plt.figure(figsize=(10, 6))
plt.plot(years, vessel_quantities, marker='o', label='Vessel Quantities')
plt.plot(years, emissions, marker='o', label='CO2 Emissions')
plt.legend()
plt.title('Yearly Comparison of Vessel Quantities and CO2 Emissions')
plt.xlabel('Year')
plt.ylabel('Total Quantity / Emissions')
plt.show()
```



The graph compares the total *quantities of vessels* and *CO2 emissions* from 2022 to 2024. The x-axis represents the years, and the y-axis shows the total quantities or emissions.

Key Observations:

There is only a slight increase in the '*vessel quantity*' from 2022 to 2023. However, a good increase can be seen in 2024.

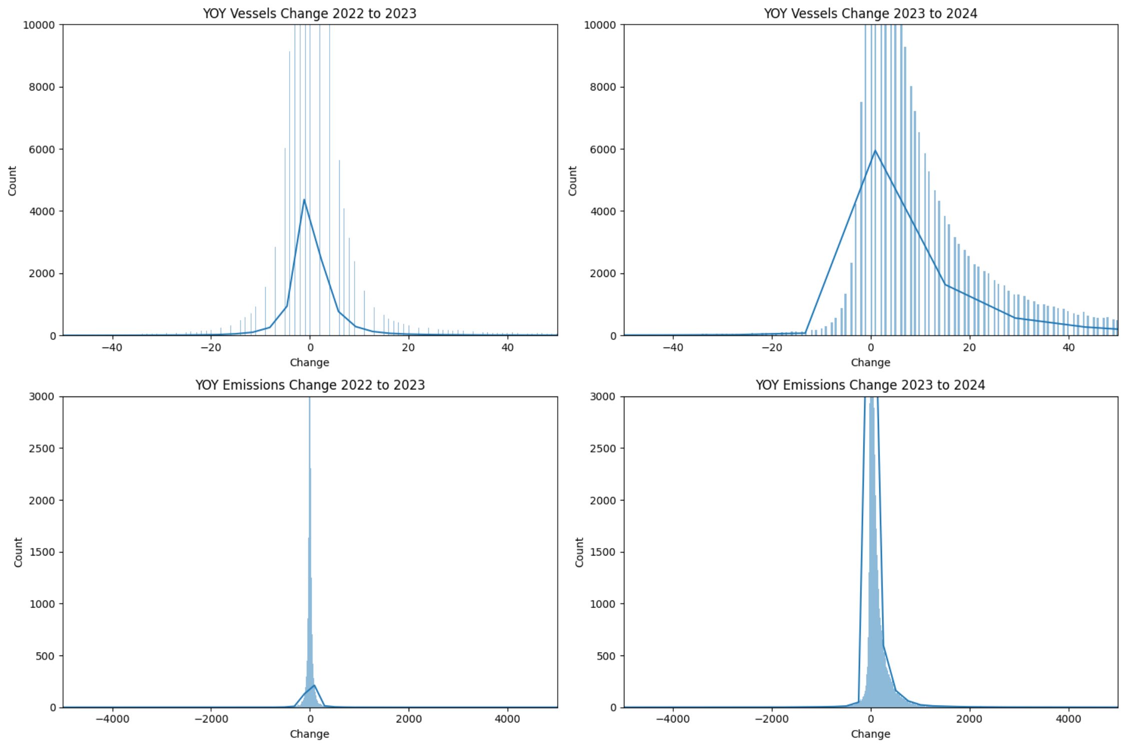
CO2 Emissions remain stable in the year 2023 and very slight increase can be observed. However, the emissions rise significantly in 2024 (as can be seen from the graph above).

Conclusion

The graph shows that while the number of vessels has a slow and steady increase, CO2 emissions remain stable at first and then dramatically increase in 2024. This indicates that even with a relatively small increase in the number of vessels, CO2 emissions have risen sharply by 2024.

[68]:
```
# Calculate year-over-year changes
df['YOY_vessels_2022_to_2023'] = df['Qty_vessels_2023'] - df['Qty_vessels_2022']
df['YOY_vessels_2023_to_2024'] = df['Qty_vessels_2024'] - df['Qty_vessels_2023']
df['YOY_emissions_2022_to_2023'] = df['c02_emissions_2023'] -
    df['c02_emissions_2022']
df['YOY_emissions_2023_to_2024'] = df['c02_emissions_2024'] -
    df['c02_emissions_2023']
```

[69]:
```
# Set up the figure and axes
fig, axs = plt.subplots(2, 2, figsize=(15, 10))

# Plot each Year-over-Year change with fixed axes
sns.histplot(df['YOY_vessels_2022_to_2023'], ax=axs[0, 0], kde=True)
axs[0, 0].set_title('YOY Vessels Change 2022 to 2023')
axs[0, 0].set_xlim(-50, 50)  # Adjust as necessary
axs[0, 0].set_ylim(0, 10000)  # Adjust as necessary

sns.histplot(df['YOY_vessels_2023_to_2024'], ax=axs[0, 1], kde=True)
axs[0, 1].set_title('YOY Vessels Change 2023 to 2024')
axs[0, 1].set_xlim(-50, 50)  # Adjust as necessary
axs[0, 1].set_ylim(0, 10000)  # Adjust as necessary

sns.histplot(df['YOY_emissions_2022_to_2023'], ax=axs[1, 0], kde=True)
axs[1, 0].set_title('YOY Emissions Change 2022 to 2023')
axs[1, 0].set_xlim(-5000, 5000)  # Adjust as necessary
axs[1, 0].set_ylim(0, 3000)  # Adjust as necessary

sns.histplot(df['YOY_emissions_2023_to_2024'], ax=axs[1, 1], kde=True)
axs[1, 1].set_title('YOY Emissions Change 2023 to 2024')
axs[1, 1].set_xlim(-5000, 5000)  # Adjust as necessary
axs[1, 1].set_ylim(0, 3000)  # Adjust as necessary

# Set common labels
for ax in axs.flat:
    ax.set_xlabel('Change')
    ax.set_ylabel('Count')

# Adjust layout
plt.tight_layout()
plt.show()
```



0.2.5 5. Outlier Detection

[70]:
```python
# Detect outliers using z-scores
from scipy.stats import zscore

# Calculate z-scores for emissions columns
df['zscore_emissions_2022'] = zscore(df['c02_emissions_2022'])
df['zscore_emissions_2023'] = zscore(df['c02_emissions_2023'])
df['zscore_emissions_2024'] = zscore(df['c02_emissions_2024'])

# Filter out outliers (z-score > 3 or < -3)
outliers = df[(df['zscore_emissions_2022'].abs() > 3) |
              (df['zscore_emissions_2023'].abs() > 3) |
              (df['zscore_emissions_2024'].abs() > 3)]

# Display outliers
print(outliers)
```

```
       geohash  Qty_vessels_2022  Qty_vessels_2023  Qty_vessels_2024  \
46       sn8f5               117               210               217
54       stpk7               370               476               512
59       k9c71               105                97               256
92       sx40w               240               119               622
```

```
117     sfq9v              625              797              393
…       …              …              …              …
271089  sp1b6               38               41               92
271105  tj4u6              123              142              188
271126  tk1zr              131              100              323
271190  tj4z3              130              113              160
271232  sfw03              298              380              227

        c02_emissions_2022  c02_emissions_2023  c02_emissions_2024  \
46              888.905814         1479.663789         1388.984855
54             4053.844392         4331.677253         3499.634991
59              789.742667          604.928939         2556.043970
92             1523.316195          737.849175         3596.780424
117            6332.217213         6703.332711         2073.388547
…                      …                  …                  …
271089         1624.748491         1500.605901         4054.325430
271105         1342.362111         1357.492263         2218.984923
271126         1162.721076          996.530860         3314.622760
271190         1589.437958          900.357394         2842.271621
271232         2894.105607         3118.869160         1289.195337

        Pctg_Emissions_2022_Vs_2023  Pctg_Emissions_2023_Vs_2024  \
46                        66.459007                    -6.128347
54                         6.853565                   -19.208316
59                       -23.401766                   322.536236
92                       -51.562967                   387.468245
117                        5.860751                   -69.069288
…                                …                            …
271089                    -7.640727                   170.179227
271105                     1.127129                    63.462068
271126                   -14.293214                   232.616168
271190                   -43.353725                   215.682599
271232                     7.766253                   -58.664655

                                                      wkt  \
46      POLYGON ((1.18652344 36.9140625, 1.18652344 36…
54      POLYGON ((32.8271484 28.8720703, 32.8271484 28…
59      POLYGON ((24.3017578 -34.6289062, 24.3017578 -…
92      POLYGON ((25.5761719 39.4628906, 25.5761719 39…
117     POLYGON ((43.1103516 12.9638672, 43.1103516 13…
…                                                    …
271089  POLYGON ((2.54882812 39.4189453, 2.54882812 39…
271105  POLYGON ((48.9550781 28.8720703, 48.9550781 28…
271126  POLYGON ((59.0185547 23.7744141, 59.0185547 23…
271190  POLYGON ((48.9111328 29.3994141, 48.9111328 29…
271232  POLYGON ((42.2314453 14.1064453, 42.2314453 14…

                                                 geometry  \
```

```
46      POLYGON ((1.18652344 36.9140625, 1.18652344 36…
54      POLYGON ((32.8271484 28.8720703, 32.8271484 28…
59      POLYGON ((24.3017578 -34.6289062, 24.3017578 -…
92      POLYGON ((25.5761719 39.4628906, 25.5761719 39…
117     POLYGON ((43.1103516 12.9638672, 43.1103516 13…
…                                                   …
271089  POLYGON ((2.54882812 39.4189453, 2.54882812 39…
271105  POLYGON ((48.9550781 28.8720703, 48.9550781 28…
271126  POLYGON ((59.0185547 23.7744141, 59.0185547 23…
271190  POLYGON ((48.9111328 29.3994141, 48.9111328 29…
271232  POLYGON ((42.2314453 14.1064453, 42.2314453 14…

        YOY_vessels_2022_to_2023  YOY_vessels_2023_to_2024  \
46                            93                         7
54                           106                        36
59                            -8                       159
92                          -121                       503
117                          172                      -404
…                            …                        …
271089                         3                        51
271105                        19                        46
271126                       -31                       223
271190                       -17                        47
271232                        82                      -153

        YOY_emissions_2022_to_2023  YOY_emissions_2023_to_2024  \
46                      590.757975                  -90.678934
54                      277.832861                 -832.042262
59                     -184.813728                 1951.115031
92                     -785.467021                 2858.931249
117                     371.115498                -4629.944164
…                              …                          …
271089                 -124.142590                 2553.719529
271105                   15.130152                  861.492660
271126                 -166.190216                 2318.091900
271190                 -689.080564                 1941.914227
271232                  224.763553                -1829.673823

        zscore_emissions_2022  zscore_emissions_2023  zscore_emissions_2024
46                   2.066899               3.569151               1.978090
54                  10.546032              11.050565               5.578762
59                   1.801233               1.274543               3.969039
92                   3.766537               1.623219               5.744488
117                 16.649981              17.271901               3.145651
…                         …                      …                      …
271089               4.038283               3.624086               6.525039
271105               3.281746               3.248670               3.394032
271126               2.800473               2.301794               5.263140
```

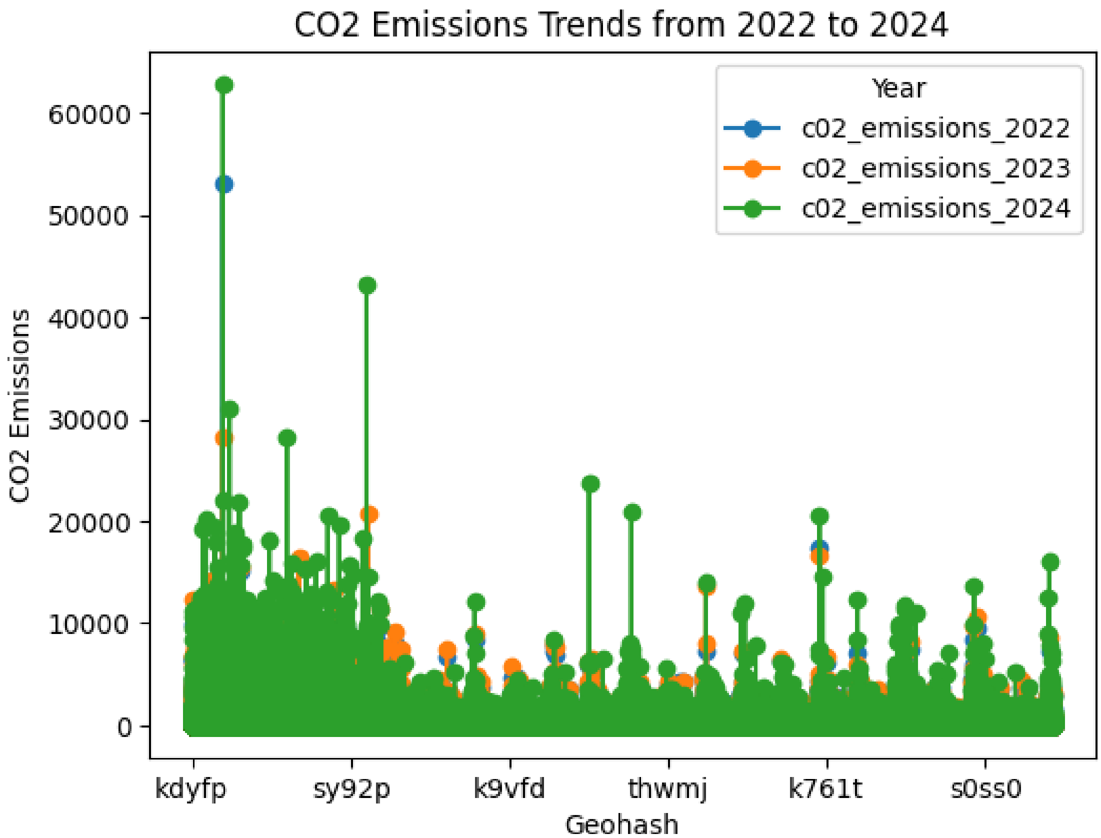
```
271190               3.943683               2.049512               4.457331
271232               7.438995               7.869121               1.807854

[5739 rows x 18 columns]
```

0.2.6 6. Trend Analysis Over Time

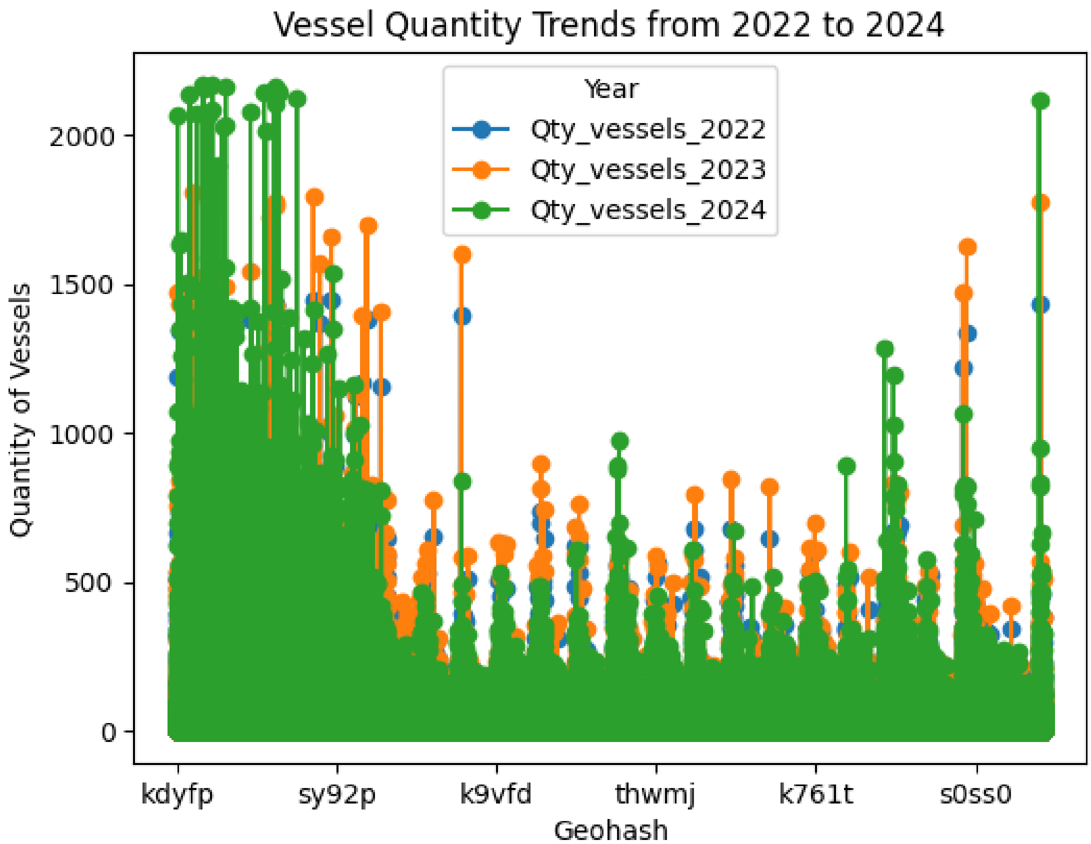
Instead of calculating rolling averages, I visualized the trend over time for emissions and vessel quantities.

[71]:
```python
# Plot trends over years
plt.figure(figsize=(14, 8))
df.set_index('geohash')[['c02_emissions_2022', 'c02_emissions_2023',␣
↪'c02_emissions_2024']].plot(marker='o')
plt.title('CO2 Emissions Trends from 2022 to 2024')
plt.xlabel('Geohash')
plt.ylabel('CO2 Emissions')
plt.legend(title='Year')
plt.show()

plt.figure(figsize=(14, 8))
df.set_index('geohash')[['Qty_vessels_2022', 'Qty_vessels_2023',␣
↪'Qty_vessels_2024']].plot(marker='o')
plt.title('Vessel Quantity Trends from 2022 to 2024')
plt.xlabel('Geohash')
plt.ylabel('Quantity of Vessels')
plt.legend(title='Year')
plt.show()
```

```
<Figure size 1400x800 with 0 Axes>
```




```
<Figure size 1400x800 with 0 Axes>
```



The above scatter plots represent the CO2 emissions and number of vessels across different locations (denoted by geohash codes) for the years 2022, 2023, and 2024.

The key points include:

Geohash Codes: Locations are represented by unique geohash codes on the x-axis.

CO2 Emissions: The y-axis shows the amount of CO2 emissions and number of vessels respectively.

Yearly Data: Each color represents a different year (2022 in blue, 2023 in orange, and 2024 in green).

Figure 1: CO2 Emissions Trends from 2022 to 2024

There is a noticeable increase in CO2 emissions in 2024 across many geohash codes compared to the previous years, indicating a rising trend in emissions.

Figure 2: Vessel Quantity Trends from 2022 to 2024

There is a visible increase in the quantity of vessels in 2024 across many geohash codes compared to the previous years, suggesting a growing number of vessels over time.

0.2.7 7. Clustering Analysis

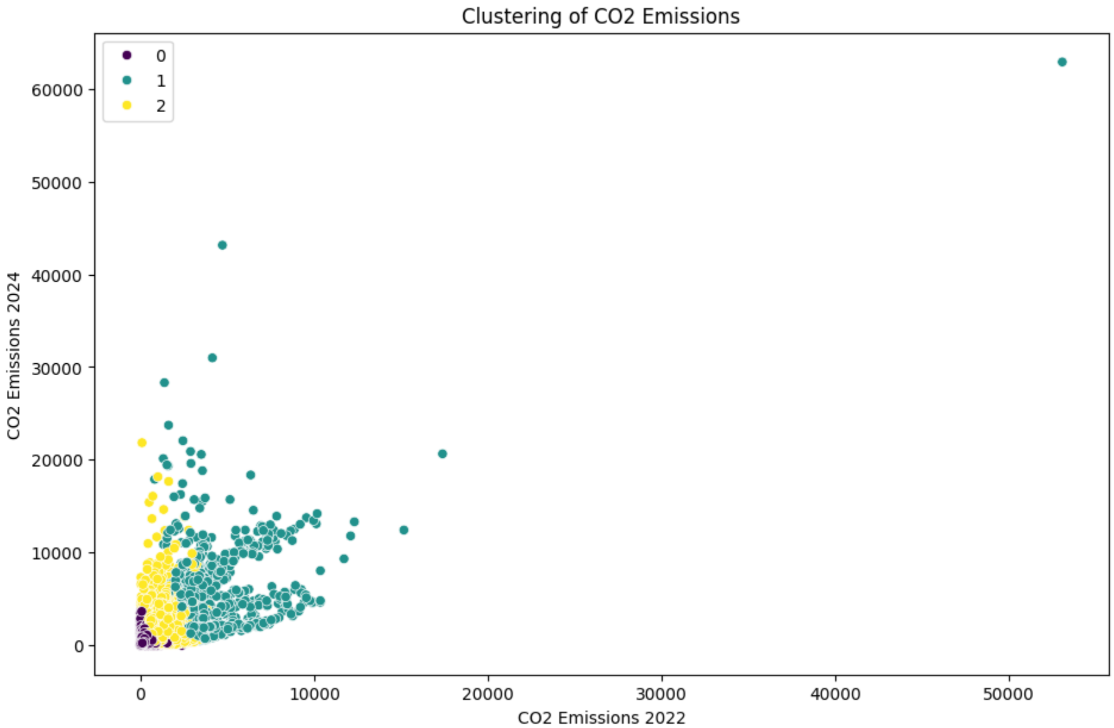
Clustering was executed using three clusters. The results indicate that the clustering process was effective with these three clusters. The clusters have distinct boundaries and minimal overlaps, demonstrating that the clustering was performed correctly.

[72]:
```python
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans

# Prepare the data for clustering
X = df[['Qty_vessels_2022', 'Qty_vessels_2023', 'Qty_vessels_2024',
                'c02_emissions_2022', 'c02_emissions_2023',␣
↪'c02_emissions_2024']]

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

[73]:
```python
# Perform clustering
kmeans = KMeans(n_clusters=3, random_state=0)
df['cluster'] = kmeans.fit_predict(X_scaled)

# Add cluster labels to the GeoDataFrame
gdf['cluster'] = df['cluster']
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:1416:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
```

[81]:
```python
# Plot clustering results
plt.figure(figsize=(11, 7))
sns.scatterplot(x='c02_emissions_2022', y='c02_emissions_2024', hue='cluster',␣
↪data=df, palette='viridis')
plt.title('Clustering of CO2 Emissions')
plt.xlabel('CO2 Emissions 2022')
plt.ylabel('CO2 Emissions 2024')
plt.legend()
plt.show()
```



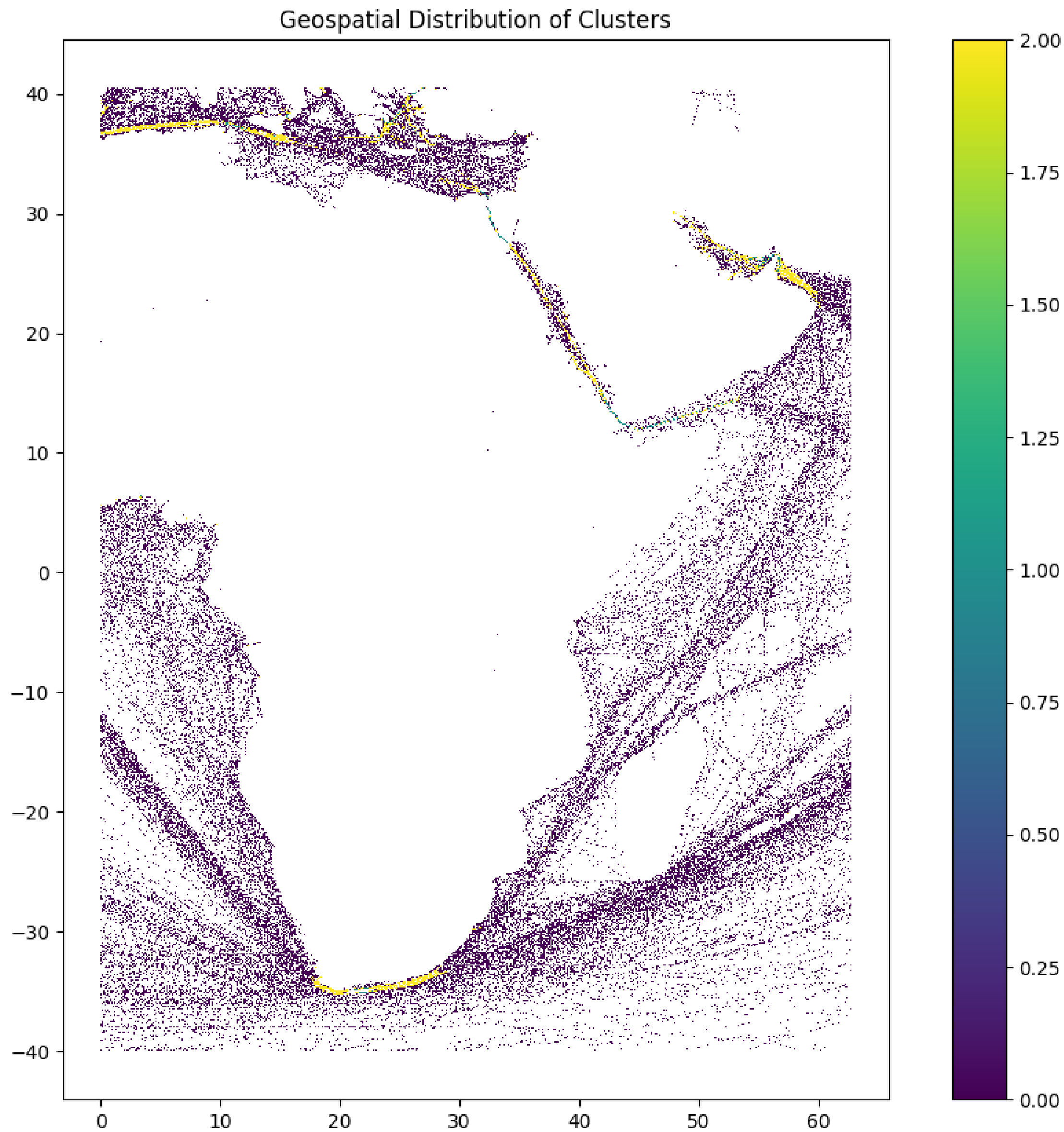
The above graph shows the cluster 2 (yellow) which had a smaller range of CO2 emissions in 2022 had a much bigger range (spread) in 2024.

Cluster 1 (blue) represents points with a high emission and a greater number of vessels.

0.2.8 8. Geospatial Analysis (Enhanced)

Enhance the geospatial analysis by visualizing clusters or using a more detailed map.

[75]:
```
import geopandas as gpd

# Convert POLYGON column to GeoDataFrame
gdf = gpd.GeoDataFrame(df, geometry=gpd.GeoSeries.from_wkt(df['wkt']))

# Plot the geospatial data with clusters
fig, ax = plt.subplots(1, 1, figsize=(12, 10))
gdf.plot(column='cluster', ax=ax, legend=True, cmap='viridis')
plt.title('Geospatial Distribution of Clusters')
plt.show()
```



The image displays a geospatial distribution of clusters, likely from a clustering analysis of vessel data and CO2 emissions.

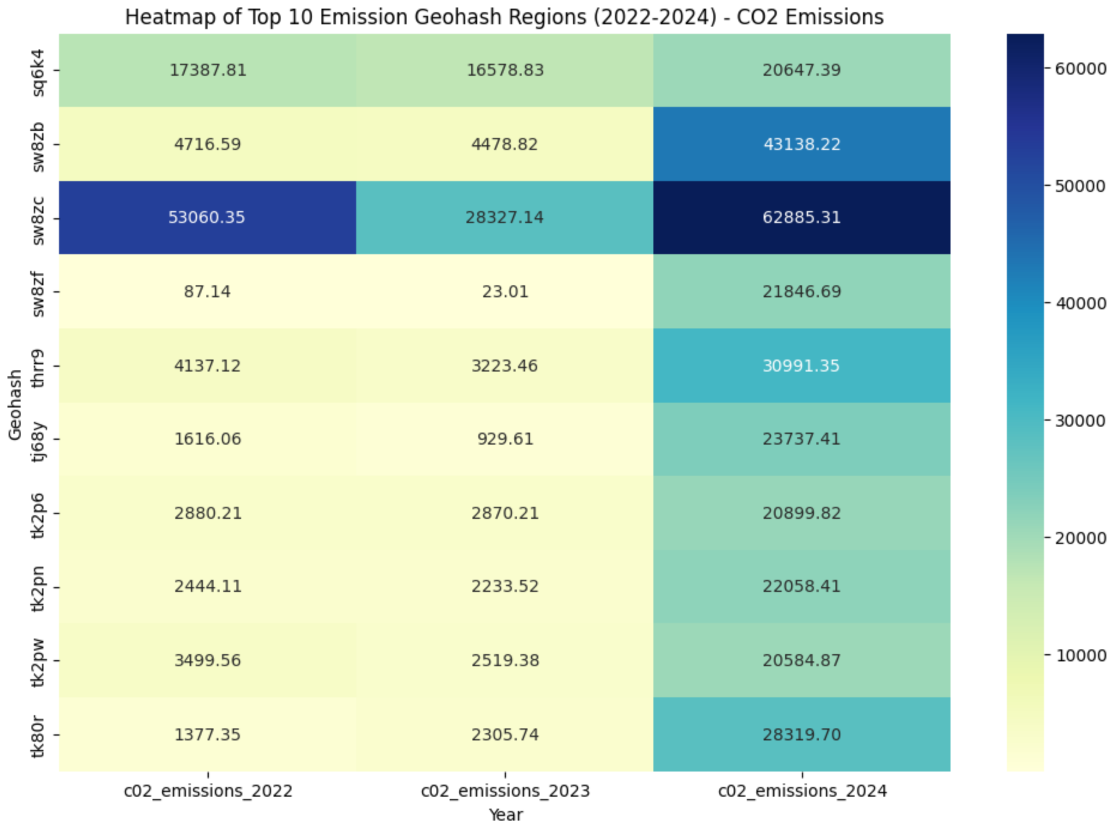
Cluster Visualization: The data points are color-coded based on cluster assignments, with a color gradient ranging from purple (0) to yellow (2).

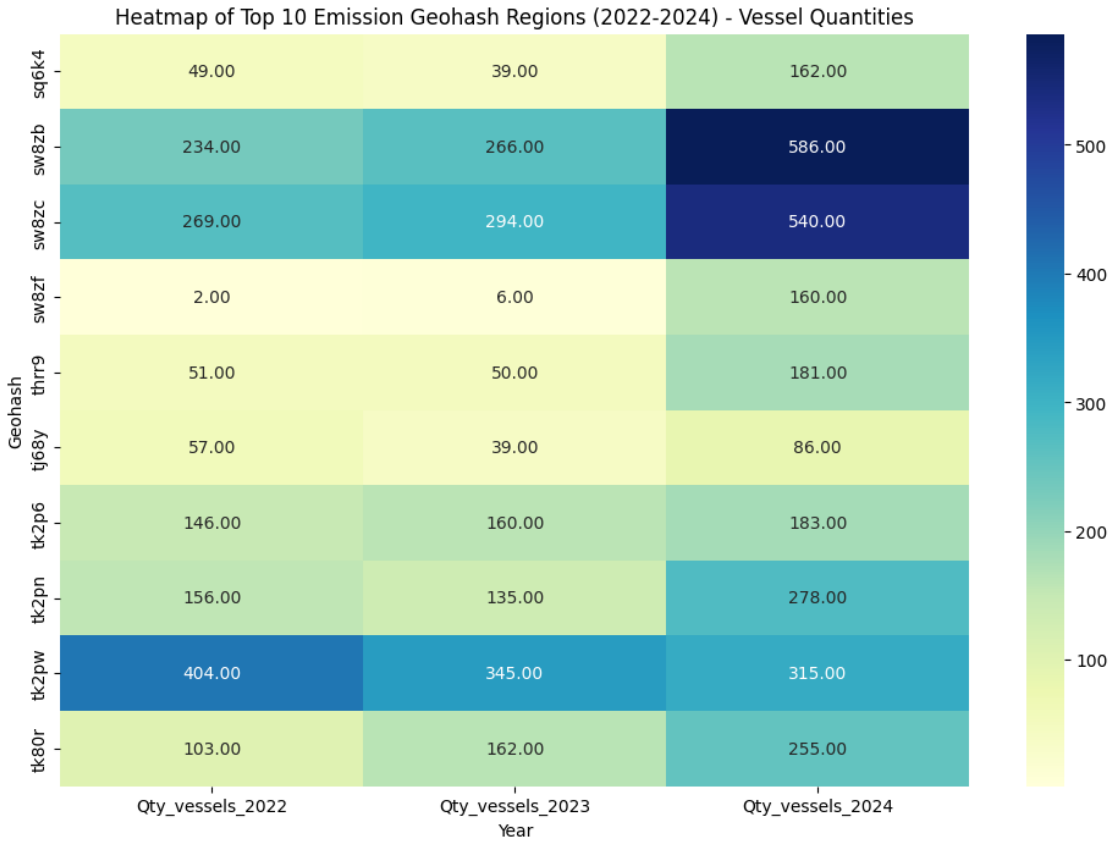
Yellow areas indicate higher cluster values (2), while purple areas represent lower values (0).

Density: The density of points and the color intensity provide insights into the concentration of data points within each cluster. Higher density regions (yellow) suggest areas with significant maritime activity or higher emissions.

Clustering Effectiveness: The distinct color boundaries and minimal overlaps between different clusters suggest that the clustering has effectively grouped the data points based on their similarities.

[82]:
```
# Find top 10 geohashes by total emissions in 2024
top_10 = df.nlargest(10, 'c02_emissions_2024')

# Melt the DataFrame for heatmap plotting (CO2 Emissions)
melted_top_10_emissions = top_10.melt(id_vars='geohash',␣
↪value_vars=['c02_emissions_2022', 'c02_emissions_2023',␣
↪'c02_emissions_2024'],
                                      var_name='Year', value_name='CO2␣
↪Emissions')

# Create a pivot table for CO2 Emissions
pivot_top_10_emissions = melted_top_10_emissions.pivot(index='geohash',␣
↪columns='Year', values='CO2 Emissions')

# Plot heatmap for top 10 CO2 Emissions
plt.figure(figsize=(12, 8))
sns.heatmap(pivot_top_10_emissions, cmap="YlGnBu", annot=True, fmt='.2f')
plt.title('Heatmap of Top 10 Emission Geohash Regions (2022-2024) - CO2␣
↪Emissions')
plt.xlabel('Year')
plt.ylabel('Geohash')
plt.show()

# Melt the DataFrame for heatmap plotting (Vessel Quantities)
melted_top_10_vessels = top_10.melt(id_vars='geohash',␣
↪value_vars=['Qty_vessels_2022', 'Qty_vessels_2023', 'Qty_vessels_2024'],
                                    var_name='Year', value_name='Vessel␣
↪Quantities')

# Create a pivot table for Vessel Quantities
pivot_top_10_vessels = melted_top_10_vessels.pivot(index='geohash',␣
↪columns='Year', values='Vessel Quantities')

# Plot heatmap for top 10 Vessel Quantities
plt.figure(figsize=(12, 8))
sns.heatmap(pivot_top_10_vessels, cmap="YlGnBu", annot=True, fmt='.2f')
plt.title('Heatmap of Top 10 Emission Geohash Regions (2022-2024) - Vessel␣
↪Quantities')
plt.xlabel('Year')
plt.ylabel('Geohash')
plt.show()
```


Heatmap of Top 10 Emission Geohash Regions (2022-2024) - CO2 Emissions
Geohash
sq6k4
sw8zb
sw8zc
sw8zf
thrr9
tj68y
tk2p6
tk2pn
tk2pw
tk80r
17387.81
16578.83
20647.39
4716.59
4478.82
43138.22
53060.35
28327.14
62885.31
87.14
23.01
21846.69
4137.12
3223.46
30991.35
1616.06
929.61
23737.41
2880.21
2870.21
20899.82
2444.11
2233.52
22058.41
3499.56
2519.38
20584.87
1377.35
2305.74
28319.70
c02_emissions_2022
c02_emissions_2023
c02_emissions_2024
Year
60000
50000
40000
30000
20000
10000



The above two heatmaps show the CO2 emissions and the number of vessels in the top 10 emission geohash regions from 2022 to 2024.

Key Observations:

Top Emitting Regions

Regions like sw8zc and sq6k4 have significantly higher emissions (dark blue). sw8zc has the highest emissions in all three years, especially in 2022 and 2024.

High Vessel Regions

sw8zb and sw8zc have the highest number of vessels, especially in 2024, where sw8zb reaches 586 vessels and sw8zc has 540 vessels.

tk2pw also shows a relatively high number of vessels across the years, particularly in 2022.

Increasing Trends:

Regions like sw8zb, sw8zc, and tk2p6 show an increasing trend in the number of vessels over the years.

For example, sw8zb increases from 234 vessels in 2022 to 586 vessels in 2024.

Stable or Decreasing Trends:

Some regions, like sq6k4, thrnp, and tk2pn, show more stability or slight fluctuations in vessel numbers.

tk2pn starts with 156 vessels in 2022, decreases to 135 vessels in 2023, and then rises again to 278 vessels in 2024.

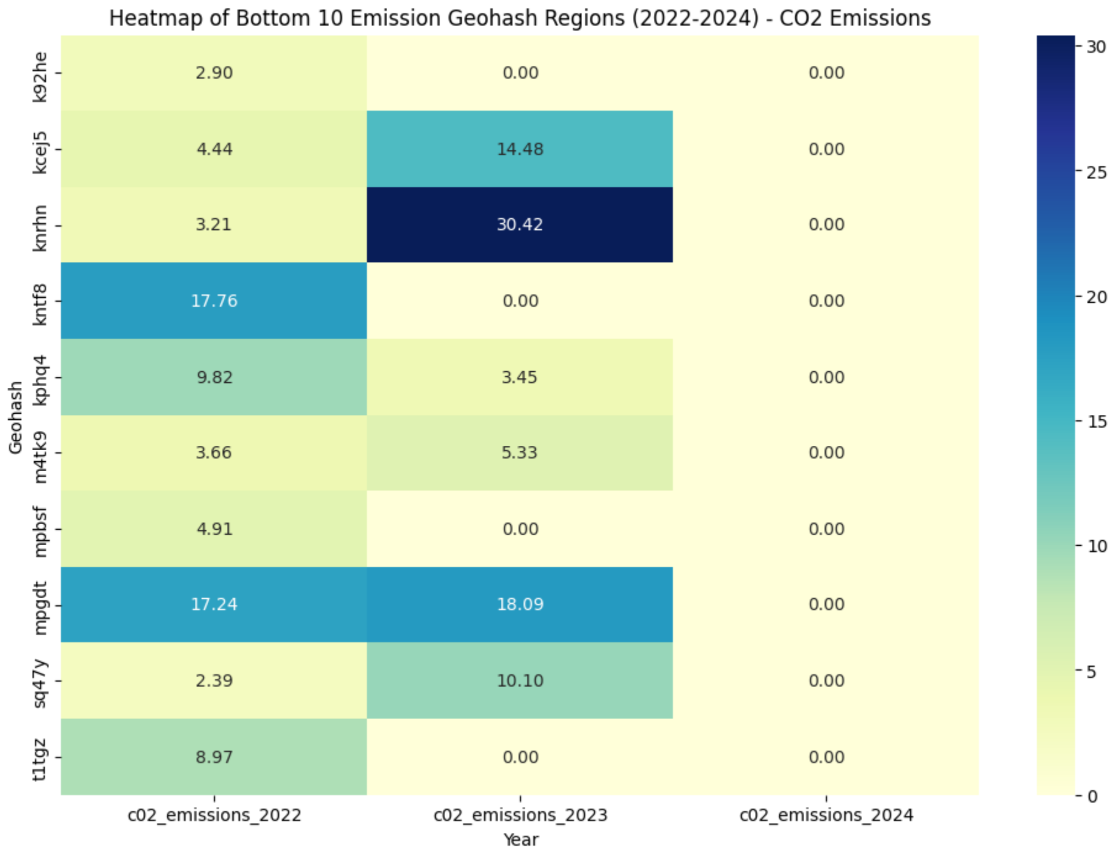
Low Vessel Regions:

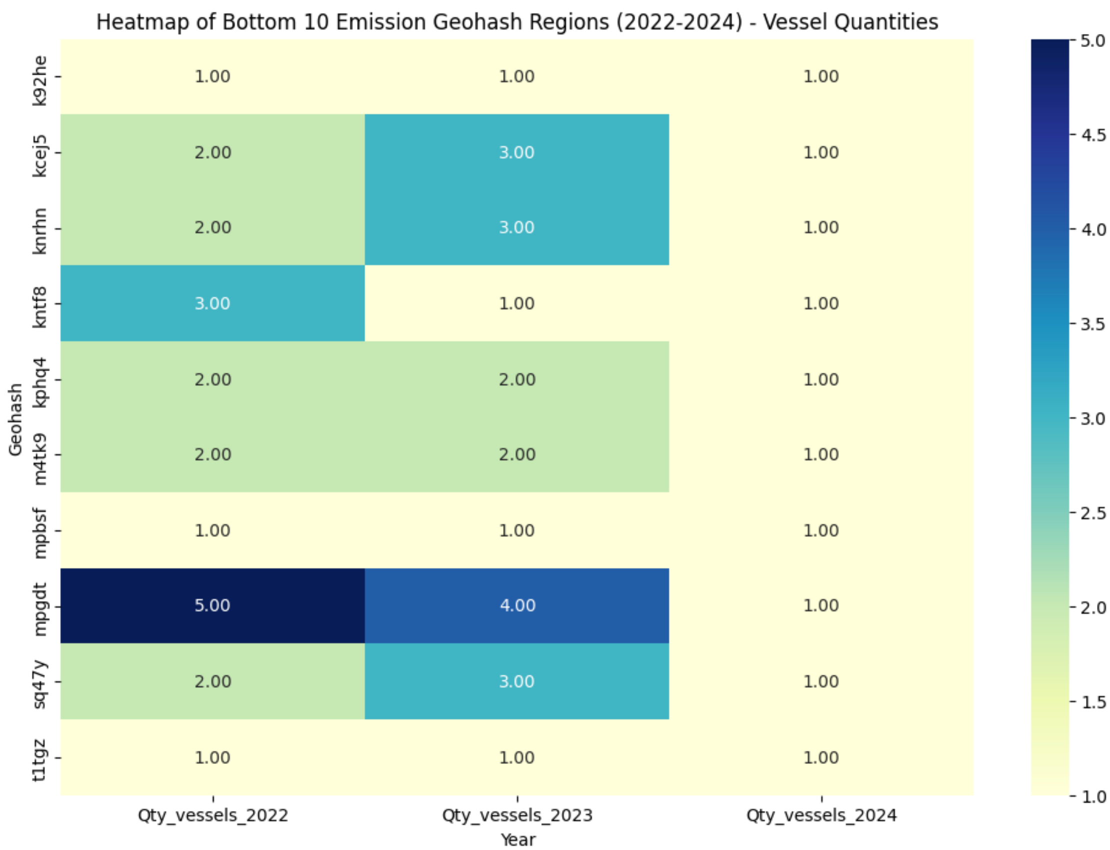
sw8zf consistently shows the lowest number of vessels, with only 2 in 2022 and gradually increasing to 160 in 2024.

[83]:

```
bottom_10 = df.nsmallest(10, 'c02_emissions_2024')

# Melt the DataFrame for heatmap plotting (CO2 Emissions)
melted_bottom_10_emissions = bottom_10.melt(id_vars='geohash',␣
↪value_vars=['c02_emissions_2022', 'c02_emissions_2023',␣
↪'c02_emissions_2024'],
                                            var_name='Year', value_name='CO2␣
↪Emissions')

# Create a pivot table for CO2 Emissions
pivot_bottom_10_emissions = melted_bottom_10_emissions.pivot(index='geohash',␣
↪columns='Year', values='CO2 Emissions')

# Plot heatmap for bottom 10 CO2 Emissions
plt.figure(figsize=(12, 8))
sns.heatmap(pivot_bottom_10_emissions, cmap="YlGnBu", annot=True, fmt='.2f')
plt.title('Heatmap of Bottom 10 Emission Geohash Regions (2022-2024) - CO2␣
↪Emissions')
plt.xlabel('Year')
plt.ylabel('Geohash')
plt.show()

# Melt the DataFrame for heatmap plotting (Vessel Quantities)
melted_bottom_10_vessels = bottom_10.melt(id_vars='geohash',␣
↪value_vars=['Qty_vessels_2022', 'Qty_vessels_2023', 'Qty_vessels_2024'],
                                          var_name='Year', value_name='Vessel␣
↪Quantities')

# Create a pivot table for Vessel Quantities
pivot_bottom_10_vessels = melted_bottom_10_vessels.pivot(index='geohash',␣
↪columns='Year', values='Vessel Quantities')

# Plot heatmap for bottom 10 Vessel Quantities
plt.figure(figsize=(12, 8))
sns.heatmap(pivot_bottom_10_vessels, cmap="YlGnBu", annot=True, fmt='.2f')
plt.title('Heatmap of Bottom 10 Emission Geohash Regions (2022-2024) - Vessel␣
↪Quantities')
```

```python
plt.xlabel('Year')
plt.ylabel('Geohash')
plt.show()
```



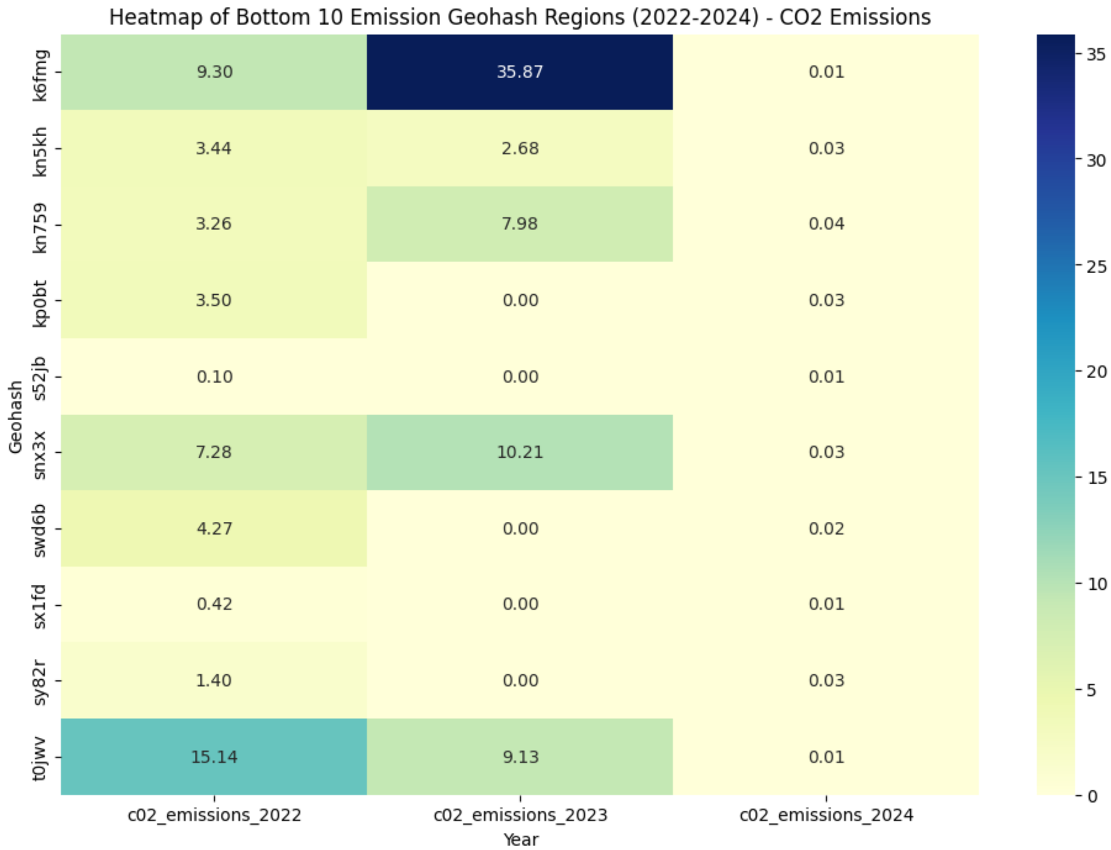


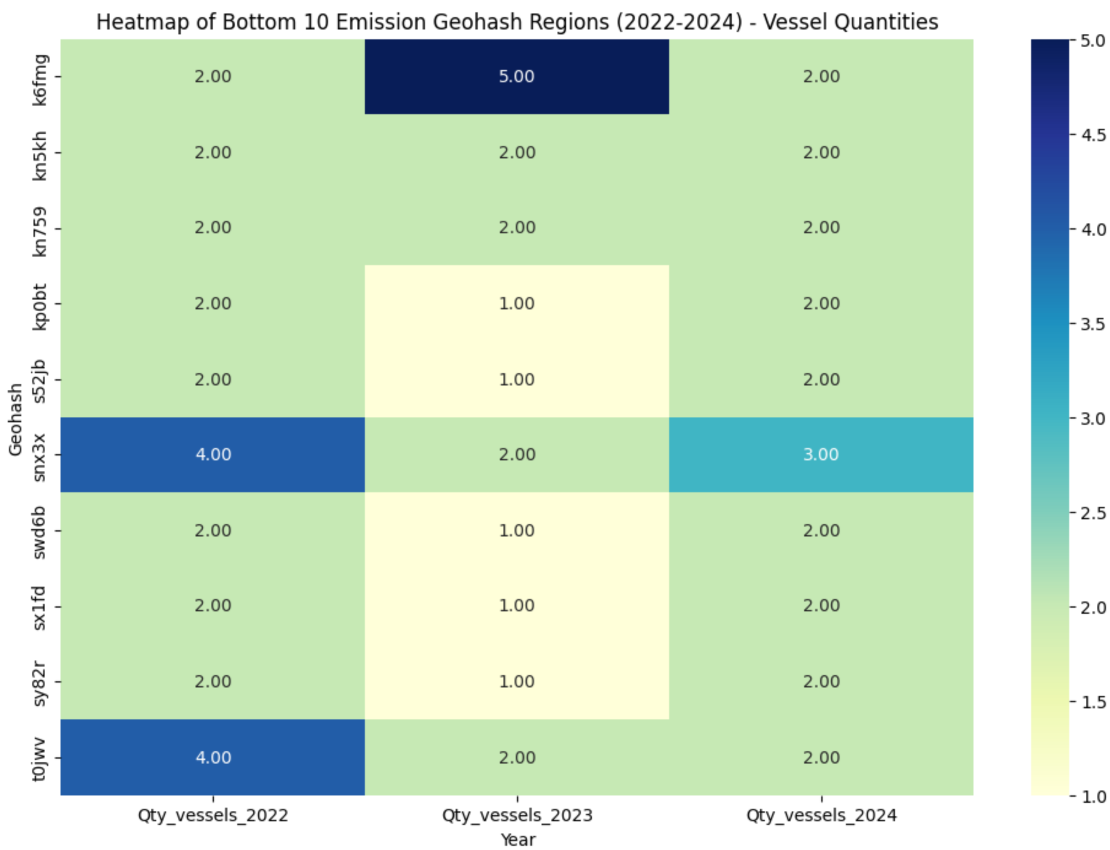
Low Emitting Regions:

Regions like k9zhe, kcej5, and knrhn have very low emissions (light yellow). These regions consistently show minimal or zero emissions across all three years.

[84]:
```
# Filter out geohashes where CO2 emissions are 0 in 2024
filtered_df = df[df['c02_emissions_2024'] > 0]

# Find bottom 10 geohashes by total emissions in 2024 from the filtered␣
↪DataFrame
bottom_10 = filtered_df.nsmallest(10, 'c02_emissions_2024')

# Melt the DataFrame for heatmap plotting (CO2 Emissions)
melted_bottom_10_emissions = bottom_10.melt(id_vars='geohash',␣
↪value_vars=['c02_emissions_2022', 'c02_emissions_2023',␣
↪'c02_emissions_2024'],
                                            var_name='Year', value_name='CO2␣
↪Emissions')

# Create a pivot table for CO2 Emissions
```

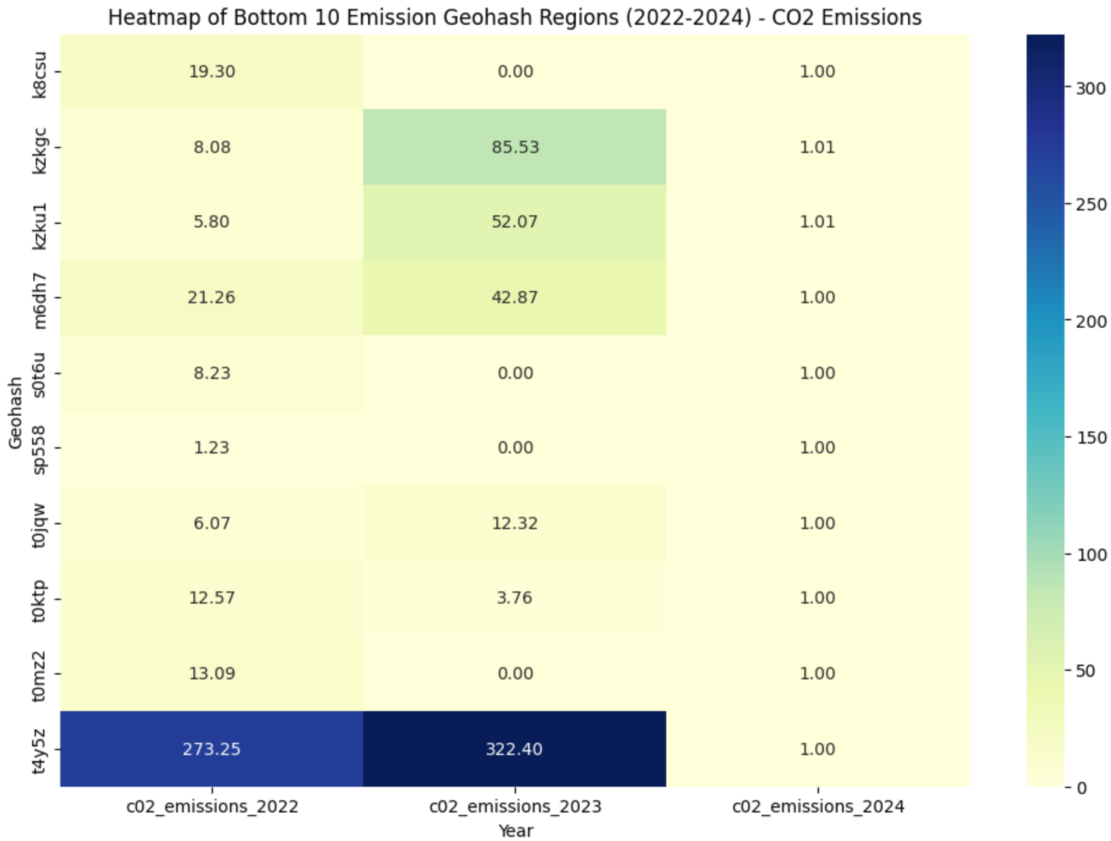
```
pivot_bottom_10_emissions = melted_bottom_10_emissions.pivot(index='geohash',␣
↪columns='Year', values='CO2 Emissions')

# Plot heatmap for bottom 10 CO2 Emissions
plt.figure(figsize=(12, 8))
sns.heatmap(pivot_bottom_10_emissions, cmap="YlGnBu", annot=True, fmt='.2f')
plt.title('Heatmap of Bottom 10 Emission Geohash Regions (2022-2024) - CO2␣
↪Emissions')
plt.xlabel('Year')
plt.ylabel('Geohash')
plt.show()

# Melt the DataFrame for heatmap plotting (Vessel Quantities)
melted_bottom_10_vessels = bottom_10.melt(id_vars='geohash',␣
↪value_vars=['Qty_vessels_2022', 'Qty_vessels_2023', 'Qty_vessels_2024'],
                                          var_name='Year', value_name='Vessel␣
↪Quantities')

# Create a pivot table for Vessel Quantities
pivot_bottom_10_vessels = melted_bottom_10_vessels.pivot(index='geohash',␣
↪columns='Year', values='Vessel Quantities')

# Plot heatmap for bottom 10 Vessel Quantities
plt.figure(figsize=(12, 8))
sns.heatmap(pivot_bottom_10_vessels, cmap="YlGnBu", annot=True, fmt='.2f')
plt.title('Heatmap of Bottom 10 Emission Geohash Regions (2022-2024) - Vessel␣
↪Quantities')
plt.xlabel('Year')
plt.ylabel('Geohash')
plt.show()
```

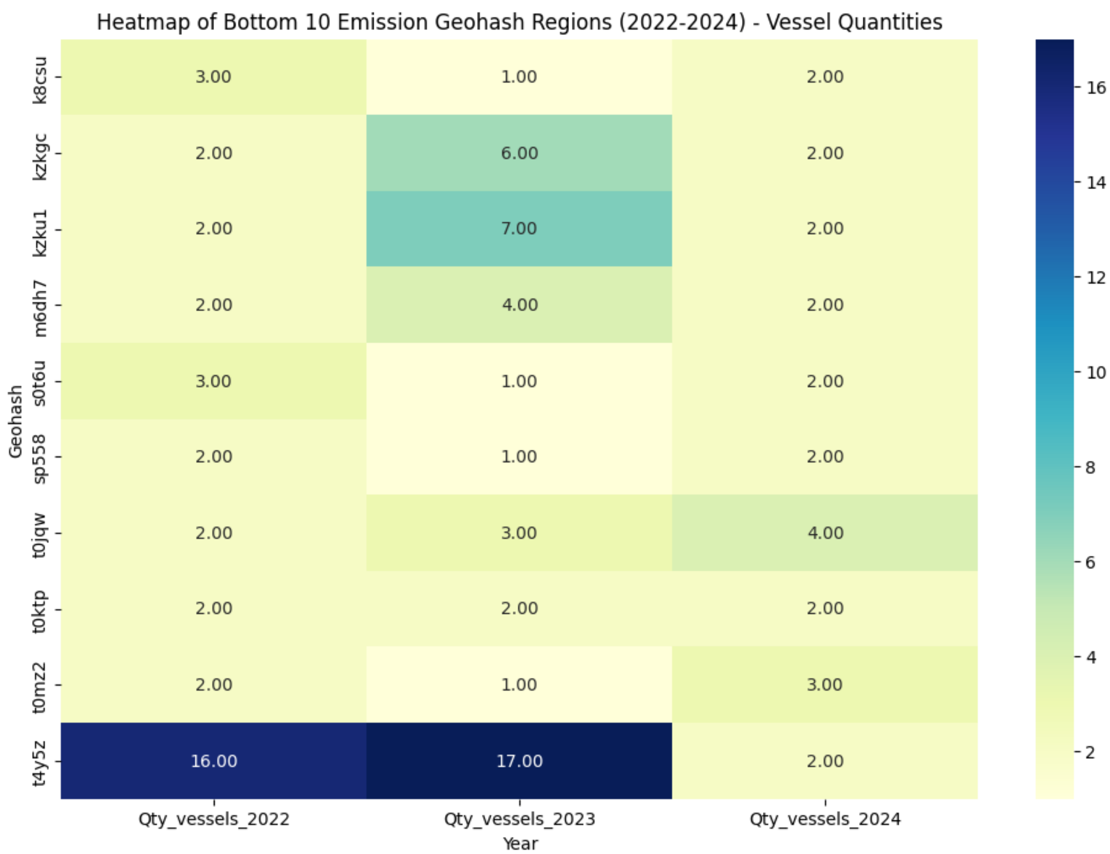

Heatmap of Bottom 10 Emission Geohash Regions (2022-2024) - CO2 Emissions
Geohash
k6fmg
kn5kh
kn759
kp0bt
s52jb
snx3x
swd6b
sx1fd
sy82r
t0jwv
9.30
35.87
0.01
3.44
2.68
0.03
3.26
7.98
0.04
3.50
0.00
0.03
0.10
0.00
0.01
7.28
10.21
0.03
4.27
0.00
0.02
0.42
0.00
0.01
1.40
0.00
0.03
15.14
9.13
0.01
c02_emissions_2022
c02_emissions_2023
c02_emissions_2024
Year
35
30
25
20
15
10
5
0



[86]:
```python
# Filter out geohashes where CO2 emissions are   1 in 2024
filtered_df = df[df['c02_emissions_2024'] > 1]

# Find bottom 10 geohashes by total emissions in 2024 from the filtered DataFrame
bottom_10 = filtered_df.nsmallest(10, 'c02_emissions_2024')

# Melt the DataFrame for heatmap plotting (CO2 Emissions)
melted_bottom_10_emissions = bottom_10.melt(id_vars='geohash', value_vars=['c02_emissions_2022', 'c02_emissions_2023', 'c02_emissions_2024'],
                                            var_name='Year', value_name='CO2 Emissions')

# Create a pivot table for CO2 Emissions
pivot_bottom_10_emissions = melted_bottom_10_emissions.pivot(index='geohash', columns='Year', values='CO2 Emissions')

# Plot heatmap for bottom 10 CO2 Emissions
plt.figure(figsize=(12, 8))
```

```
sns.heatmap(pivot_bottom_10_emissions, cmap="YlGnBu", annot=True, fmt='.2f')
plt.title('Heatmap of Bottom 10 Emission Geohash Regions (2022-2024) - CO2␣
↪Emissions')
plt.xlabel('Year')
plt.ylabel('Geohash')
plt.show()

# Melt the DataFrame for heatmap plotting (Vessel Quantities)
melted_bottom_10_vessels = bottom_10.melt(id_vars='geohash',␣
↪value_vars=['Qty_vessels_2022', 'Qty_vessels_2023', 'Qty_vessels_2024'],
                                          var_name='Year', value_name='Vessel␣
↪Quantities')

# Create a pivot table for Vessel Quantities
pivot_bottom_10_vessels = melted_bottom_10_vessels.pivot(index='geohash',␣
↪columns='Year', values='Vessel Quantities')

# Plot heatmap for bottom 10 Vessel Quantities
plt.figure(figsize=(12, 8))
sns.heatmap(pivot_bottom_10_vessels, cmap="YlGnBu", annot=True, fmt='.2f')
plt.title('Heatmap of Bottom 10 Emission Geohash Regions (2022-2024) - Vessel␣
↪Quantities')
plt.xlabel('Year')
plt.ylabel('Geohash')
plt.show()
```


Heatmap of Bottom 10 Emission Geohash Regions (2022-2024) - CO2 Emissions
Geohash
k8csu
kzkgc
kzku1
m6dh7
s0t6u
sp558
t0jqw
t0ktp
t0mz2
t4y5z
19.30
0.00
1.00
8.08
85.53
1.01
5.80
52.07
1.01
21.26
42.87
1.00
8.23
0.00
1.00
1.23
0.00
1.00
6.07
12.32
1.00
12.57
3.76
1.00
13.09
0.00
1.00
273.25
322.40
1.00
c02_emissions_2022
c02_emissions_2023
c02_emissions_2024
Year
300
250
200
150
100
50
0


Heatmap of Bottom 10 Emission Geohash Regions (2022-2024) - Vessel Quantities
Geohash
k8csu
kzkgc
kzku1
m6dh7
s0t6u
sp558
t0jqw
t0ktp
t0mz2
t4y5z
Qty_vessels_2022
Qty_vessels_2023
Qty_vessels_2024
Year
3.00
1.00
2.00
2.00
6.00
2.00
2.00
7.00
2.00
2.00
4.00
2.00
3.00
1.00
2.00
2.00
1.00
2.00
2.00
3.00
4.00
2.00
2.00
2.00
2.00
1.00
3.00
16.00
17.00
2.00
16
14
12
10
8
6
4
2